

17. Мельник Э.В., Горелова Г.В. Исследование эффекта выравнивания вычислительной нагрузки процессорных устройств в высоконадежных распределенных информационно-управляющих системах. Материалы 7-й Всероссийской научно-практ. конф. «Перспективные системы и задачи управления». – Таганрог: Изд-во ТТИ ЮФУ, 2012. – С. 316-326.
18. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач. – М.: Наука, 1982. – С. 9-15.

Статью рекомендовал к опубликованию д.т.н. В.П. Карелин.

**Горелова Галина Викторовна** – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: g.v.gorelova@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634394264; кафедра государственного и муниципального управления; д.т.н.; профессор.

**Мельник Эдуард Владиславович** – НИИ Многопроцессорных вычислительных систем им. акад. А.В. Каляева, (НИИ МВС), ЮФУ, Таганрог; e-mail: evm@mvs.tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634311865, 88634622088; к.т.н.; лаборатория НИИ МВС; зав. лабораторией.

**Gorelova Galina Viktorovna** – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: g.v.gorelova@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634394264; the department of state and municipal management; dr. of eng. sc.; professor.

**Melnik Eduard Vladislavovich** – Institute of Multiprocessor systems to them. Acad. Kalyaev, SFU, Taganrog; e-mail: evm@mvs.tsure.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phones: +78634311865, +78634622088; cand. of eng. sc.; laboratory institute of multiprocessor systems; head of laboratory.

УДК Д-1239, И46

**О.Л. Бухвалов, В.И. Городецкий, О.В. Карсаев, Г.И. Кудрявцев,  
В.В. Самойлов**

### **РАСПРЕДЕЛЕННАЯ КООРДИНАЦИЯ В ПРОИЗВОДСТВЕННЫХ В2В-СЕТЯХ**

*Описывается эффективный распределенный алгоритм координации расписаний выполнения потока взаимосвязанных заказов в В2В-сетях предприятий с ограниченными производственными мощностями при заданных ограничениях на допустимые периоды выполнения заказов с учетом требований сохранения конфиденциальности информации узлов сети. Предлагается его реализация в архитектуре многоагентных систем. Алгоритм демонстрируется на примере задачи межцеховой координации расписаний большого предприятия, в котором цехи сами ответственны как за составление своих локальных расписаний выполнения множества заказов, так и за их координацию с аналогичными расписаниями других цехов.*

*Распределенная координация локальных расписаний; сохранение конфиденциальности; агент; многоагентная архитектура.*

**O.L. Bukhvalov, V.I. Gorodetsky, O.V. Karsaev, G.I. Kudryavtsev, V.V. Samoylov**

### **DISTRIBUTED COORDINATION IN B2B PRODUCTION NETWORKS**

*The paper concern is privacy preserving distributed production scheduling in B2B-networks when several organizations of limited production powers are involved in concurrent fulfillment of multiple interconnected dynamically incoming order flow. The paper presents a privacy preserv-*

*ing distributed coordination algorithm implemented on the basis of multi-agent architecture. To validate the solution, a case study dealing with coordination of local schedules of several shops of a Russian production plant is used.*

*Distributed production planning; resource scheduling; privacy; multi-agent architecture; industrial implementation.*

**Введение.** Актуальность исследований в области Business-to-Business (B2B)-сетей как в России, так и за рубежом, обусловлена возрастающей глобализацией бизнеса, усилением конкуренции, а также усилением роли веб-сервисов в бизнесе, в частности, роли облачных веб-сервисов. Эти исследования стимулируются также возрастающей ролью компаний малого и среднего бизнеса, которые занимают существенную часть в общем объеме современного бизнеса.

В области B2B-сетей можно выделить три группы ключевых задач. Первая из них – это разработка программной платформы для поддержки взаимодействия предприятий, формирующих B2B-сеть. Такая платформа должна обеспечить информационную и программную совместимость (interoperability) бизнес-процессов различных компаний сети, а также их эффективное и безопасное взаимодействие при кооперации для совместного производства продукции (enterprise cooperation) в рамках B2B-сети. В Европе базовые принципы построения платформы для поддержки B2B-сетей сформулированы в документе [3], подготовленном международной организацией FInES Cluster ([www.fines-cluster.eu](http://www.fines-cluster.eu)). В рамках аналогичной задачи российский стандарт ГОСТ Р ИСО 14813-1-2011 [7] предусматривает четкую иерархическую структуризацию веб-сервисов, поддерживающих функционирование B2B-сетей, стандартное использование которых должно обеспечиваться упомянутой выше платформой.

Вторая группа задач имеет целью создание инструментальных средств поддержки процессов разработки компонент информационного и программного обеспечения конкретной компании, которое дает ей возможность присоединяться к B2B-сети и вести в ее рамках свой конкретный бизнес.

Наконец, третья группа задач имеет целью разработку конкретных алгоритмов координации планирования и синхронизации расписаний выполнения общего бизнес-процесса различными компаниями, которые формируют B2B-сеть.

Данная работа имеет целью представить некоторые результаты по третьей из названных выше задач, а именно теоретические и экспериментальные результаты исследований в области координации и синхронизации бизнес-процессов в B2B-сети производственной логистики. Организация кооперативного производства множества продуктов множеством различных предприятий автоматически требует решения еще одной, не менее важной и сложной задачи, а именно задачи сохранения конфиденциальной информации. Обычно компании не хотят раскрывать своим партнерам по сети информацию коммерческого характера об особенностях своей технологии, станочном парке и структуре управления и др. Требование конфиденциальности накладывает ряд жестких требований к алгоритмам координации производства в сети предприятий. Например, конфиденциальность исключает использование централизованных хранилищ данных и методов координации. Поэтому координация планов и расписаний производства предприятий производственной сети может выполняться только в *распределенном* режиме при минимуме информации, которой обмениваются партнеры сети.

В данной работе описывается *распределенный алгоритм координации расписаний* выпуска продукции предприятиями производственной B2B-сети с учетом требований конфиденциальности. Предлагается его реализация в архитектуре многоагентных систем (МАС), которая позволяет естественным образом учесть эти требования. В качестве демонстрационного примера рассматривается задача меж-

цеховой координации одного большого предприятия, в котором цехи являются самостоятельными подразделениями и сами планируют свой производственный процесс. Алгоритм обеспечивает сохранение конфиденциальной информации, он не требует участия центральных звеньев в процессе поиска решения, а также не требует обмена информацией о состоянии цеховых ресурсов и деталях технологических процессов отдельных цехов.

Работа организована следующим образом. В разд. 2 представляется постановка задачи распределенной координации расписаний в производственной В2В-сети. Разд. 3 содержит описание разработанного алгоритма координации локальных расписаний производства в узлах сети с сохранением конфиденциальности. В разд. 4 описывается МАС-архитектура программного прототипа, реализующего алгоритм. В разд. 5 кратко резюмируются результаты экспериментального исследования алгоритма распределенной координации, реализованного как МАС-приложение. В заключении обсуждаются результаты работы и планы дальнейших исследований по рассматриваемой проблематике.

### 1. Постановка задачи оптимизации использования ресурсов В2В-сети.

Разработанный алгоритм распределенной координации расписаний узлов В2В производственной сети базируется на ранее разработанном алгоритме составления расписаний использования ресурсов отдельного цеха [2]. Алгоритм координации использует его в качестве локального алгоритма составления расписаний и обобщает его на задачу распределенной координации. В основу программной реализации алгоритма распределенной координации расписаний узлов В2В производственной сети положена концепция МАС.

В постановке задачи рассматривается множество узлов В2В-сети  $\{Dep_i\}_{i=1}^{N_{Dep}}$ , которые участвуют в совместном выполнении множества заказов  $\{O_j\}_{j=1}^{N_{Order}}$ , из которых, по крайней мере, некоторое подмножество таково, что в их исполнении участвуют несколько различных узлов. Под заказом понимается некоторое задание на изготовление устройства, возможно, сложного и, возможно, в каком-то количестве экземпляров. Заметим, что в общем случае символу  $Dep_i$  может соответствовать не обязательно производственное предприятие. Это может быть произвольная компания или виртуальная организация, участвующая в реализации некоторого бизнес-процесса, компоненты которого могут на практике соответствовать технологическим процессам по доставке некоторого множества продуктов потребителям, их хранению и т.п.

Специфической особенностью узла  $Dep_i$  В2В-сети является то, что он обладает некоторым множеством ресурсов и технологией их использования для выполнения тех или иных работ. Другими словами, узел сети может выполнять некоторые заказы или их части и планирует их выполнение самостоятельно. Полагается, что каждый узел стремится оптимизировать некоторый локальный показатель качества или несколько таких показателей в более общем случае. Кроме того, каждому заказу, который должен быть выполнен подмножеством узлов В2В-сети, поставлены в соответствие два времени:  $\{T_j^S\}_{j=1}^{N_{Order}}$  – самое раннее время, когда выполнение заказа может быть начато, и  $\{T_j^F\}_{j=1}^{N_{Order}}$  – самый поздний срок, к которому он должен быть выполнен полностью. Подчеркнем, что эти ограничения относятся к заказу в целом, а не к его частям. Они задают интервал календарного времени (времена начала и окончания), в течение которого заказ может выполняться и должен быть закончен.

В постановке задачи полагается, что любой заказ может быть разбит на части, каждая из которых должна выполняться предопределенным узлом В2В-сети<sup>1</sup>. Множество частей (подзаказов) одного заказа частично упорядочено по отношению следования в технологическом процессе производства. Поэтому некоторые подзаказы, в общем случае, могут выполняться параллельно разными узлами сети. В свою очередь, каждый подзаказ выполняется в соответствии с технологией, которая описывается частично упорядоченным множеством простых действий (технологических операций)  $\{a_i\}_{i=1}^{N(O)}$ . Важно, что как именно выполняется каждый отдельный подзаказ некоторым узлом сети (технология, используемые ресурсы и расписание на множестве операций) другим узлам В2В-сети неизвестно. Им может быть доступна только информация о том, какой узел сети выполняет тот или иной подзаказ, а также когда выполнение подзаказа может быть им начато и когда оно предположительно может быть закончено.

Каждая операция  $a \in \{a_i\}_{i=1}^{N(O)}$  подзаказа заказа  $O$ , выполняемая узлом  $Dep_i$ , требует использования некоторых ресурсов одного из следующих типов [4]:

- ◆ *возобновляемые* ресурсы: после использования для одной операции они могут вновь использоваться для другой, но в каждый момент времени они могут использоваться только одной из операций; примерами таких ресурсов в производственной логистике являются станки, рабочие и др.
- ◆ *невозобновляемые* ресурсы, которые потребляются технологическими операциями; для них указывается общий доступный объем; примерами являются деньги, материалы, комплектующие изделия и др.;
- ◆ *ресурсы, имеющие двухсторонние ограничения*; для них указывается максимальный объем, доступный на различных интервалах времени (например, фонд заработной платы в различные периоды времени).

В рассматриваемой постановке задачи полагается, что ограниченными являются только возобновляемые ресурсы. В рамках приложения, которое далее используется для экспериментальной проверки алгоритма распределенной координации, полагается, что каждый станок закреплен за рабочим, и рабочий имеет квалификацию (разряд), который принимает дискретное значение из множества  $\{1, \dots, b\}$ . Таким образом, процедура составления расписания использования ресурсов принимает во внимание ограничения на множество допустимых исполнителей операций с учетом их квалификации и доступности во времени. Последнее определяется длительностью рабочего дня, возможной недоступностью исполнителей по причине болезней, отпусков и т.п. В конкретном приложении могут иметься свои ограничения, не рассматриваемые в общей постановке задачи.

С учетом введенных понятий и ограничений *задача распределенной координации* формулируется следующим образом:

Построить расписания использования ограниченных ресурсов каждого узла В2В-сети, выполняющей заданное множество взаимосвязанных подзаказов на фиксированном интервале времени, которые удовлетворяют:

- (1) ограничениям на допустимые временные интервалы выполнения заказов,
- (2) технологическим ограничениям на частичный порядок выполнения операций подзаказов и на частичный порядок выполнения самих подзаказов,
- (3) требованию конфиденциальности информации, которой узлы сети обмениваются, и
- (4) оптимизируют использование ресурсов (возможно, по многим критериям) каждого узла сети.

<sup>1</sup> В данной работе не рассматривается, каким именно образом заказ разбивается на части и как определяются исполнители этих частей. Эта задача не относится к теме работы.

Примерами оптимизируемых показателей качества являются объем валовой или товарной продукции, доход каждого узла в терминах некоторой функции полезности, среднее время задержки выполнения заказов и др.

Сделаем одно существенное замечание по поводу рассматриваемой постановки задачи. Может создаться впечатление, что она имеет целью только локальную оптимизацию производственных расписаний узлов сети. Однако это не так. Поскольку имеются ограничения на начало исполнения каждого заказа, а также на самый поздний срок его завершения, то все узлы сети оказываются еще и в рамках глобального ограничения. Оно ведет к тому, что локальные расписания являются зависимыми; они должны координироваться так, чтобы удовлетворить указанному глобальному требованию.

**2. Локальный алгоритм оптимизации производственного расписания.** Постановка задачи оптимизации использования ограниченных ресурсов узла и алгоритм ее решения с использованием МАС-архитектуры рассмотрены в [1, 2]. В архитектуре его программной реализации используются следующие типы агентов:

- ◆ *агент узла* (один в каждом узле): управляет распределением ресурсов узла по работам подзаказов, назначенных ему на горизонте планирования;
- ◆ *агент подзаказа* (один для каждого подзаказа): ответственен за выполнение технологических и временных ограничений, обусловленных подзаказом;
- ◆ *агент ресурса* (соответствует конкретному рабочему): ответственен за совместимость типов операций, поручаемых рабочему, и его специализации, а также за выполнение ограничений на его доступность во времени.

Опуская детали, работу локального алгоритма можно описать четырьмя последовательными фазами, которые реализуют стандартный *Протокол контрактных сетей (Contract Net Protocol, CNP)* [6]:

А. *Агент подзаказа* определяет для подзаказа список операций, которые могут выполняться далее с учетом текущего состояния его исполнения и частичным порядком на множестве операций. Этот список операций далее называется *фронт-том планирования*. Он пересылается *Агенту узла*.

В. *Агент узла* исполняет CNP-протокол: выбирает из списка *одну* операцию, используя эвристический механизм<sup>2</sup>, и посылает запрос агентам ресурсов узла, в какой период они могут выполнить предлагаемую операцию.

С. *Агент ресурса* решает, может он или не может выполнить предложенную операцию, и если может, то возвращает агенту узла свое предложение, в котором указывает период исполнения операции и качество ее исполнения.

Д. *Агент узла* выбирает наилучшее предложение, используя *Политику выбора победителя* и информирует об этом "победившего" *Агента ресурса*, который резервирует для выполнения этой операции ресурс и слот времени.

Описанный алгоритм иницируется каждый раз, когда в В2В-сеть поступает новый заказ. При этом пересчитывается расписание всех тех операций, которые исполняются в момент поступления заказа или их исполнение еще не было начато. Алгоритм работает очень быстро, поэтому пересчет расписаний в реальном времени не вызывает задержек в работе сети. Например, для того, чтобы составить расписание для 2000 заказов на 4–5 месяцев при общем количестве операций до 40 000 при штатном составе цеха 80–90 рабочих и станочном парке до 150 станков, требуется не более 10 мин времени персонального компьютера средней мощности.

<sup>2</sup> Множество эвристик и правила их использования при выборе очередной операции для включения в расписание называется "политикой выбора". *Политика* рассматривается как компонента повторного использования, настраиваемая в контексте ее использования.

**3. Распределенная координация.** Алгоритм распределенной координации локальных расписаний узлов В2В-сети строится на основе обмена метаданными между ее узлами в процессе составления ими локальных расписаний. В МАС-архитектуре его программной реализации задействованы агенты локального уровня, описанные в предыдущем разделе, а также агенты метауровня, ответственные за координацию. В ней на локальном и метауровнях используются агенты следующих классов:

- ◆ *Агент узла (Department Agent, DA):* ответственен за решение локальных задач (см. предыдущий раздел), но дополнительно он взаимодействует с аналогичными агентами других узлов, решая совместно с ними задачу координации. Этот агент выполняет, таким образом, отдельные функции метауровня.
- ◆ *Агент ресурса (Resource Agent, RA):* участвует в выполнении CNP-протокола так, как было описано в предыдущем разделе.
- ◆ *Локальный агент заказа (Local order agents, LOA):* выполняет те же функции, что и аналогичный агент, описанный в предыдущем разделе. Зоной его ответственности является локальная часть заказа, выполняемая в узле.
- ◆ *Агент заказа (Order-agent, OA).* Он владеет информацией о заказе в целом: о самом раннем допустимом времени начала его исполнения и о самом позднем времени его завершения. Этот агент является агентом метауровня.

Таким образом, в этой архитектуре, дополнительно к локальным агентам составления расписаний введены еще агенты заказов (по числу заказов), а также расширены функции агентов узлов, которые управляют не только составлением локальных расписаний, но и решают задачи их координации. Важно отметить, что в этой архитектуре информация о ресурсах и технологиях узлов доступна только агентам соответствующих узлов, что решает задачу конфиденциальности.

Итак, в алгоритме распределенной координации участвуют *DA-агенты* и *OA-агенты*. Запуск алгоритма распределенной координации (иначе-протокола координации) осуществляется так называемыми *смежными событиями*. Каждое такое событие генерируется тогда, когда заканчивается составление расписания исполнения части заказа, выполняемой в некотором узле, а продолжение выполнения этого заказа предполагается в другом узле (или узлах), который в соответствии с планом выполнения заказа является смежным (непосредственно следующим) для узла, генерирующего смежное событие.

Еще одно дополнительное понятие, которое используется в алгоритме координации, это *маркер фронта планирования*. Маркером называется метка, которая указывает, что некоторая операция принадлежит фронту планирования.

Распределенная координация имеет целью координацию локальных расписаний узлов. Идея координации близка к той идее, которая используется в алгоритме локального составления расписаний. На текущей фазе исполнения локального алгоритма каждая операция фронта планирования метится маркером фронта, который снимается с операции после того, как она включена в расписание.

Понятие смежного события используется для обнаружения моментов, когда необходимо инициировать алгоритм координации. Оно является базовым понятием алгоритма распределенной координации, поскольку оно реализует его операционную семантику. Каждый раз, когда смежное событие генерируется, управление процессом координации передается агенту соответствующего заказа, который выполняет свою часть алгоритма координации расписаний.

Дальнейшее выполнение заказа может включать в себя выполнение одного или параллельно нескольких подзаказов другими узлами сети. Последователя также может и не быть, если операция, инициировавшая смежное событие, является последней в подзаказе, а этот подзаказ является последней частью заказа.

Каждое смежное событие ассоциировано с именем агента, которому информация об окончании выполнения одной из финальных операций подзаказа должна быть сообщена. Этот агент может быть агентом класса *LOA* или *DA*. Когда такая операция завершается, генерируется столько смежных событий, сколько последователей у этой операции имеется в других узлах в соответствии с технологией исполнения заказа. Таким образом, с каждым смежным событием связано два факта, существенные для того узла, в котором должно продолжаться выполнение заказа: (1) окончание предшествующей операции и (2) разрешение включить во фронт планирования последователей тех узлов сети, в которых по технологии выполнение заказа должно продолжаться. Если смежное событие отвечает последней операции подзаказа, то информация об этом посылается *DA-агенту*. Он инициирует окончание процесса составления локального расписания и запускает такой алгоритм в узле-последователе, выполняя тем самым функции элемента потока управления процессом координации расписаний.

Рассмотрим непосредственно алгоритм распределенной координации. Базовая идея алгоритма демонстрируется на рис. 1<sup>3</sup>. Координация выполняется путем переговоров двух *DA-агентов*: *DA-агента* узла сети, в котором завершилось выполнение порученной ему части заказа, и *DA-агента* узла, в котором выполнение заказа должно быть продолжено. Обозначим *DA-агента* первого узла символом  $DA^-$ , а *DA-агента* второго узла символом  $DA^+$ .

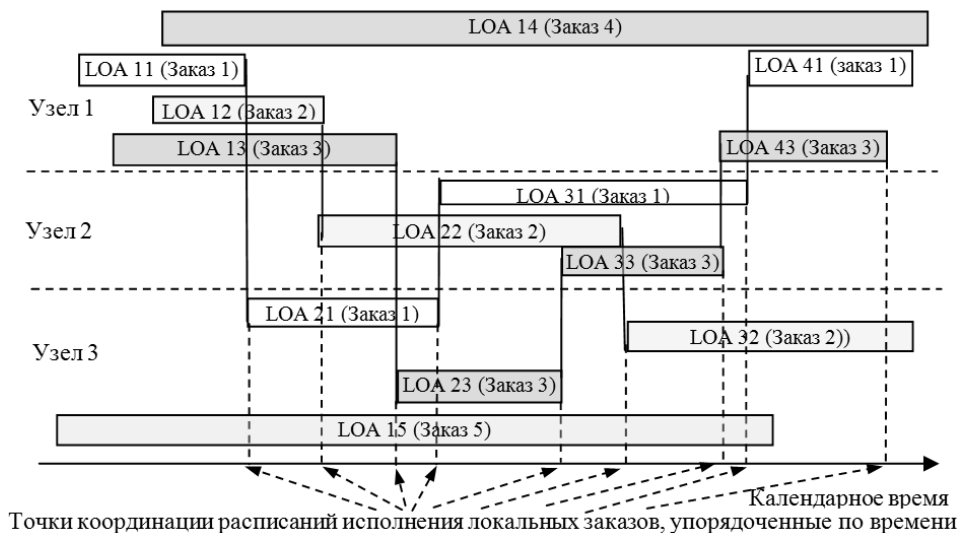


Рис. 1. Пояснение базовой идеи алгоритма распределенной координации.  
В примере общее число заказов – 5, число узлов – 3

В момент начала координации операции узла-последователя, продолжающие технологии выполнения заказа, должны быть включены во фронт планирования узла, когда  $DA^+$ -агент получил смежное событие от  $DA^-$ -агента. Получив это событие,  $DA^+$ -агент приостанавливает процесс составления локального расписания до тех пор, пока не закончится его взаимодействие с агентом  $DA^-$ .

<sup>3</sup> В описании алгоритма опущены несущественные детали, обусловленные спецификой конкретного приложения.

В начале процесса составления расписания все операции локальных подзаказов узла, которые принадлежат фронту планирования, помечены соответствующим маркером. Локальный цикл составления расписания продолжается до тех пор, пока имеется хотя бы одна операция, помеченная маркером. В случае, когда таких операций нет, локальный цикл работы алгоритма переходит в режим ожидания и стартует тогда, когда в узле появляются операции, помеченные маркером.

Иначе говоря, алгоритм организован так, что глобальный фронт планирования (объединение таких фронтов по множеству узлов) продвигается в нем и по временной шкале, и по структурированному множеству операций заказов, исполняемых сетью. Когда заканчивается планирование некоторого заказа, то в конечном узле выполняются некоторые заключительные операции.

Многоагентная архитектура программной реализации алгоритма представлена на рис. 3. Псевдокод алгоритма распределенной координации локальных расписаний узлов можно найти по адресу <http://ips-logistic.com/pub/inter-dep-coordination.pdf>.

Сделаем несколько замечаний относительно алгоритма в целом. Каждому заказу поставлено в соответствие значение самого позднего допустимого срока его выполнения. Как уже упоминалось, это ограничение относится к заказу в целом и является обязательным для принятия во внимание всеми узлами, задействованными в его исполнении. В описании алгоритма явно не дано информации о том, каким образом достигается учет самого позднего срока выполнения заказа всеми его соисполнителями. До начала работы описанной части алгоритма исполняется простой распределенный алгоритм, который вычисляет самое раннее и самое позднее допустимое время выполнения заказов и такие же интервалы времени вычисляются для каждой операции. В процессе составления расписаний эти границы являются управляемыми атрибутами, значения которых выбираются на основе локальных политик, реализующих процедуры многокритериального выбора. Пример подобной политики упомянут в разделе 3. Эти политики управляют решениями, принимаемыми в процессе распределенной координации таким образом, чтобы либо обеспечить удовлетворение ограничений на максимально допустимый срок исполнения заказа, либо минимизировать некоторую функцию, аргументами которой являются временные превышения этих сроков, например, их среднее значение.

**4. Программная реализация и экспериментальные исследования.** Дадим краткую характеристику программного прототипа, реализующего распределенную координацию локальных расписаний узлов производственной B2B-сети и некоторых результатов его экспериментального исследования.

Программная реализация алгоритма выполнена для приложения, которое решает задачу межцеховой координации расписаний выполнения заказов. Это приложение по своему масштабу соответствует индустриальному уровню. Для тестирования программы и оценки качества работы алгоритма использовались реальные данные Ижевского мотозавода "Аксион Холдинг". Приложение, решающее задачу стратегического планирования отдельного цеха на горизонте в один год, в настоящее время находится на этом заводе в опытной эксплуатации с августа 2011 г. Оно работает параллельно с ранее принятой технологией, в которой сама задача составления расписаний решается вручную. Архитектура части системы, отвечающей за локальное планирование, включает в себя детально разработанную онтологию и базу данных (учетную систему)<sup>4</sup>, а также программы, реализующие интерпретацию запросов, сформулированных в терминах понятий онтологии, в реляционной базе данных (в учетной системе предприятия).

<sup>4</sup> В некоммерческой системе, разработанной специально для экспериментальных исследований алгоритма координации, использованы некоторые компоненты системы централизованного составления расписаний, разработанные совместно с компанией "Разумные решения", г. Самара.



Экспериментальное исследование алгоритма распределенной координации работы узлов B2B-сети с целью проверки его работоспособности выполнялась путем сравнения результатов централизованного и распределенного составления расписаний на одних и тех же данных по заказам, ресурсам и технологии выполнения заказов. При этом для моделирования агентов использовалась квази-агентская библиотека *Scala Actors* [5]. Для программной реализации процесса тестирования использовались также дополнительные программные компоненты, специально разработанные для этой цели. Технология, поддерживаемая библиотекой *Scala Actors library*, которая использует понятие *акторов*, использовалась, в основном, для разработки многопоточных компонент приложения. Эта технология реализует приложение как множество программных сущностей, исполняемых в различных потоках и взаимодействующих, как и агенты, путем обмена сообщениями. Акторы моделируют агентов в МАС-приложениях. Выбор *Scala Actors*-библиотеки обусловлен тем, что она дает возможность построить простое и масштабируемое программное решение, что удобно для процесса тестирования. Например, эксперименты показали, что технология акторов способна обрабатывать до 5 000 сообщений в секунду независимо от числа акторов в системе. Язык программирования акторов имеет простой синтаксис и ряд удобных механизмов [5].

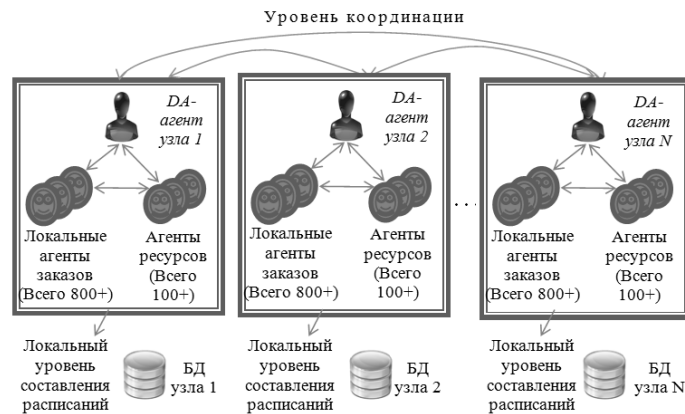


Рис. 2. Архитектура многоагентной системы распределенного составления расписаний выполнения взаимосвязанных заказов в производственной B2B-сети

Механизм распределенной координации процессов составления расписаний агентами тестировался на реальных данных инструментального цеха крупного предприятия, а именно "Ижевского мотозавода "Аксион Холдинг". Решалась задача координации работы трех цехов, выполняющих совместно заказы, которые, в свою очередь, представляются частично упорядоченным множеством простых операций. Данные, представленные в централизованной БД, были разбиты на три набора данных. Каждый набор соответствовал множеству подзаказов, которые должны выполняться в конкретном цехе, данным о технологии выполнения подзаказов в этом цехе, а также описанию ресурсов цеха. На рис. 3 показана архитектура системы тестирования, специально разработанная для оценки работоспособности, и качества алгоритма распределенной координации. В этой системе параллельно на одних и тех же исходных данных вычисляются два расписания выполнения заказов: одно расписание соответствует случаю использования централизованного алгоритма, а второе – использованию распределенной координации локальных расписаний цехов. На следующем шаге производится сравнение свойств использованных алгоритмов и качества решений, полученных с помощью каждого из них.



Рис. 3. Архитектура системы тестирования и оценки качества работы алгоритма координации локальных расписаний

Эксперименты показали, что решения, получаемые с помощью этих алгоритмов, по качеству различаются несущественно. В некоторых случаях централизованный алгоритм давал даже несколько худшее решение, однако это различие возникает за счет приближенности алгоритмов и локального характера поиска. Централизованный алгоритм работал иногда быстрее распределенного алгоритма, что частично можно рассматривать как «плату» за сохранение конфиденциальности. Однако, в основном, это естественно. Первый алгоритм оптимизирован в более ранних разработках, а в распределенный алгоритм еще предстоит оптимизировать. Кроме того, в реальном приложении распределенный алгоритм будет исполняться параллельно на серверах узлов В2В-сети, а потому – существенно быстрее. Однако он затрачивает еще дополнительное время на обмен сообщениями в процессе координации. И такой обмен может быть достаточно интенсивным. Однако это естественное свойство МАС.

**Заключение.** Экспериментальные исследования разработанного алгоритма распределенной координации, реализованного программно в архитектуре много-агентных систем, показало его хорошие качества, как в смысле вычислительной эффективности, так и в смысле качества расписания, построенного с помощью этого алгоритма. Это заключение получено на основе решения приложения практически индустриального масштаба.

Специфическим свойством разработанного алгоритма является то, что координация не требует доступа к данным или обмена данными, которые носят конфиденциальный характер. Он требует обмена только минимальной информацией о том, когда в соответствии с локально построенной частью расписания можно планировать выполнение продолжения заказа. Однако при этом принимается во внимание глобальное ограничение на самое позднее допустимое календарное время завершения заказа, и это ограничение в неявном виде оказывает влияние на аналогичный допустимый срок выполнения подзаказов каждым узлом логистической сети. Сохранение конфиденциальности информации в настоящее время является требованием первостепенной важности в связи с общей тенденцией на специализацию производств, глобализацию мест размещения предприятий, увеличивающейся ролью малых предприятий в мировом распределении функций производства товаров и услуг.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бухвалов О.Л., Вылегжанин А.С., Городецкий В.И., Карсаев О.В., Кудрявцев Г.И., Самойлов В.В. Стратегическое планирование и управление конфликтами в производственных системах с высокой динамикой заказов // Труды VII Немецко-Российской конференции по логистике и управлению в сетях поставок (DR-LOG 2012). – Санкт-Петербург, 16-19 мая 2012 г. – С. 381-388.
2. Бухвалов О.Л., Городецкий В.И., Карсаев О.В., Кудрявцев Г.И., Самойлов В.В. Производственная логистика: Стратегическое планирование, прогнозирование и управление конфликтами // Известия ЮФУ. Технические науки. – 2012. – № 3 (128). – С. 209-218.
3. Future Internet Enterprise Systems (FInES): Research Roadmap 2025. [http://cordis.europa.eu/fp7/ict/enet/documents/fines-research-roadmap-v30\\_en.pdf](http://cordis.europa.eu/fp7/ict/enet/documents/fines-research-roadmap-v30_en.pdf).
4. Sprecher A. et al. An exact algorithm for project scheduling with multiple modes // Operations-Research-Spectrum. – 1997. – Vol. 19, Issue 3. – P. 195-203.
5. Scala language, official site: <http://www.scala-lang.org/>.
6. Smith R. The contract net protocol: high-level communication and control in distributed problem solver // IEEE Transactions on Computers. – 1980. – № 29. – P. 1104-1113.
7. ГОСТ Р ИСО 14813-1-2011. <http://www.klubok.net/gost/index/51/51125.htm>.

Статью рекомендовал к опубликованию д.т.н., профессор Б.К. Гранкин.

**Бухвалов Олег Леонидович** – НПК «Интеллектуальные платформы и решения» (Сколково); e-mail: [bukhvalov@ips-logistic.com](mailto:bukhvalov@ips-logistic.com); 163015, г. Архангельск, ул. Холмогорская, 16, кв. 33; тел.: 88123233570, 89643607652; старший программист.

**Карсаев Олег Владиславович** – e-mail: [karsaev@ips-logistic.com](mailto:karsaev@ips-logistic.com); 194291, Санкт-Петербург, пр. Просвещения, 41, кв. 192; тел.: 88123233570, 88125973714; к.т.н.; генеральный директор.

**Самойлов Владимир Владимирович** – e-mail: [samoylov@ips-logistic.com](mailto:samoylov@ips-logistic.com); 197341, Санкт-Петербург, Аллея Поликарпова, 3 кв. 754; тел.: 88123233570, 88123930228; главный архитектор.

**Городецкий Владимир Иванович** – Санкт-Петербургский институт информатики и автоматизации РАН; e-mail: [gog@ias.spb.su](mailto:gog@ias.spb.su); 195349, Санкт-Петербург, пр. Сизова, 32, корп. 1, кв. 634; тел.: 88123233570, 88123011019; д.т.н.; профессор; главный научный сотрудник лаборатории интеллектуальных систем.

**Кудрявцев Геннадий Иванович** – Ижевский мотозавод «Аксион Холдинг»; e-mail: [dep115@general.adm.ru](mailto:dep115@general.adm.ru); 426000, Удмуртская республика, г. Ижевск, ул. Горького, 90; тел.: 83412783074; генеральный директор.

**Buhvalovs Oleg Leonidovich** – Senior programmer NPK «Intelligent platforms and solutions» (SKOLKOVO); e-mail: [bukhvalov@ips-logistic.com](mailto:bukhvalov@ips-logistic.com); 16, Xolmogorskaya street, ap. 33, Arxangel'sk, 163015, Russia; phones: +78123233570, +79643607652; senior programmer.

**Karsaev Oleg Vladislavovich** – e-mail: [karsaev@ips-logistic.com](mailto:karsaev@ips-logistic.com); 41, pr. Prosveshheniya, ap. 192, St. Petersburg, 194291; Russia; phones: +78123233570, +78125973714; cand. of eng. sc.; general director.

**Samoilov Vladimir Vladimirovich** – e-mail: [samoylov@ips-logistic.com](mailto:samoylov@ips-logistic.com); 3, alley Polikarpova, ap. 754, St. Petersburg, 197341, Russia; phones: +78123233570, +78123930228; chief architect.

**Gorodetsky Vladimir Ivanovich** – St. Petersburg Institute for Informatics and automation of RAS; e-mail: [gog@ias.spb.su](mailto:gog@ias.spb.su); 32/1, ap. 634, pr. Sizova, St. Petersburg, 195349, Russia; phones: +78123233570, +78123011019; dr. of eng. sc.; professor; chief researcher of intelligent systems laboratory.

**Kudryavtsev Gennady Ivanovich** – Izhevsk motozavod «Axion-Holding»; e-mail: [dep115@general.adm.ru](mailto:dep115@general.adm.ru); 90, Gorky street, Izhevsk, Udmurtia, Republic, 426000; phone: +73412783074; general director.